Tracking evolving communities in large linked networks

John Hopcroft*, Omar Khan†, Brian Kulis‡, and Bart Selman*§

*Department of Computer Science, Cornell University, Ithaca, NY 14853; †Google, Inc., Mountain View, CA 94043; and ‡Department of Computer Science, University of Texas, Austin, TX 78712

We are interested in tracking changes in large-scale data by periodically creating an agglomerative clustering and examining the evolution of clusters (communities) over time. We examine a large real-world data set: the NEC CiteSeer database, a linked network of >250,000 papers. Tracking changes over time requires a clustering algorithm that produces clusters stable under small perturbations of the input data. However, small perturbations of the CiteSeer data lead to significant changes to most of the clusters. One reason for this is that the order in which papers within communities are combined is somewhat arbitrary. However, certain subsets of papers, called natural communities, correspond to real structure in the CiteSeer database and thus appear in any clustering. By identifying the subset of clusters that remain stable under multiple clustering runs, we get the set of natural communities that we can track over time. We demonstrate that such natural communities allow us to identify emerging communities and track temporal changes in the underlying structure of our network data.

Emergent properties of large linked networks have recently become the focus of intense study. This research is driven by the increasing complexity and importance of large networks, such as the World Wide Web, the electricity grid, and large social networks that capture relationships between individuals. Real-world networks generally exhibit properties that lie somewhere in-between those of highly structured networks and purely random ones (1–4). So far, most research has focused on using static properties, such as the connectivity of the nodes in the network and the average distance between two nodes, to explain the complex structure. However, these networks generally evolve over time and so temporal characteristics are a key source of interest. Our goal in this paper is to provide techniques for the study of the evolution of large linked networks.

In our approach, we use agglomerative clusterings of the linked network. By clustering the network at different points in time, we study its temporal evolution. This approach places a new burden on the underlying clustering method. Clustering methods can be surprisingly sensitive to minor changes of the input data. For obtaining a static view of the higher-level structure of the data, such instabilities may be acceptable because the resulting hierarchy often already reveals interesting structure. However, in tracking changes over time, we need to be able to find corresponding communities in clusterings taken from the data at different points in time. If the clusterings are very sensitive to small perturbations of the input data, distinguishing between "real" changes versus "accidental" changes in the higher-level structure becomes difficult, if not impossible. In the clusterings of our linked network data, we found there are a large number of relatively random clusters that do not correspond to real community structures. These random clusters obscure the real temporal changes. Fortunately, we found that, when performing a series of agglomerative clustering runs, each run on slightly perturbed input data, one can identify a stable set of clusters that occur in a significant proportion of the clusterings. Moreover, these stable clusters appear to correspond to the true underlying community structure of the network. We refer to such stable clusters as natural communities. We use the notion of natural communities to show that we can track these natural communities effectively over time, and can therefore characterize the temporal evolution of the network.

Data Set

We used an October 2001 snapshot of the NEC CiteSeer database (5). At that time, the CiteSeer database contained the full text and bibliographies of ≈250,000 papers. These are mostly related to computer science, with a small collection covering other topics like physics, mathematics, and economics. The papers are mostly published after 1990, and the set is growing by ≈25,000 papers per year. In addition, the database contains title and author information on another 1.6 million earlier papers that are referenced by the 250,000 set but whose full text is not contained in the database.

We analyze the citation graph induced by this data set: vertices correspond to all 1.85 million papers in the database; there is a directed edge from paper A to paper B if A references B. We call the set of 250,000 papers whose full-text and bibliography are known the core of the citation graph. The papers in the core have citations to each other and to the 1.6 million earlier papers. We do not have the reference lists for the papers outside the core. So, their out-degree is 0, whereas their in-degree is at least 1. Fig. 1 gives a pictorial representation of our graph, and Table 1 contains key statistics. The out-degree (number of papers in the bibliography of a paper) of a typical node ranges from 5 to 25. The median out-degree for the core papers is 14. Interestingly, the majority of core papers are uncited (in-degree = 0). Refs. 6 and 7 describe methods for removing inaccuracies in the CiteSeer citation graph caused by the automatic generation of the graph.

The basic statistics of this graph already reveal that its structure is very different from a standard random graph. About 1 in every 100 papers receives >20 citations, 1 in every 1,000 papers has 300 citations or more, and 18 papers of the 1.85 million have >1,000 citations. This pattern is indicative of the heavy-tailed nature of the data, characterized by a power law in the in-degree (8). An interesting research question concerns the role of the highly cited papers. For example, are such nodes essential in the definition of the hidden community structure or does such structure remain even after removing high degree nodes from the graph? Also, are such nodes essential in the formation of new communities?

This paper results from the Arthur M. Sackler Colloquium of the National Academy of Sciences, "Mapping Knowledge Domains," held May 9–11, 2003, at the Arnold and Mabel Beckman Center of the National Academies of Sciences and Engineering in Irvine, CA.

§To whom correspondence should be addressed. E-mail: selman@cs.cornell.edu.

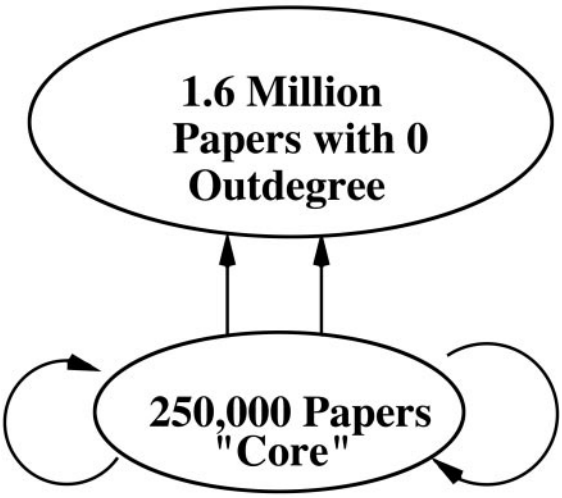

Fig. 1. Structure of NEC CiteSeer citation graph.

Instabilities and Natural Communities

Hierarchical agglomerative clustering starts with each paper in a cluster by itself. At each stage, the two "closest" clusters are merged. The process is repeated until all papers are in a single cluster. The overall process results in a "clustering tree" (referred to as a dendrogram in much of the literature), with the single paper clusters at the leaves. Each internal node corresponds to a cluster resulting from merging its two children.

Researchers have used many different distance measures. We believe that the natural community concept is valid independent of the distance measure used and thus we selected one based on cosine similarity, which is the standard similarity measure in the literature (9). With each paper p, we associate an N-dimensional reference vector r_p, where N is the total number of papers in the CiteSeer database ($N = 1.85$ million). There is a one in element i of r_p if p references paper i, otherwise the entry is 0. The similarity between two papers p and q can now be measured in terms of the cosine of the angle between the associated reference vectors, r_p and r_q. More formally, the similarity of p and q is defined to be

$$similarity(p, q) = \cos(r_p, r_q) = \frac{r_p \cdot r_q}{\|r_p\| \, \|r_q\|}, \qquad [1]$$

where $r_p \cdot r_q$ represents the inner product of r_p and r_q and $\|r_p\|$ represents the length of vector r_p. So, if two papers have no references in common, then their similarity is minimal, i.e., 0 (90° angle); two papers citing exactly the same set of papers have maximal similarity, i.e., 1 (0° angle). To get a distance measure between papers, we simply use 1− the cosine, so the distance between papers ranges from 0 to 1. When merging two papers or clusters, we represent the new cluster by the normalized sum of all of the individual papers' reference vectors, called the "centroid" of the cluster. (Our clustering method is thus a standard centroid-based agglomerative clustering technique based on cosine similarity; ref. 10.) For a cluster containing a single paper, the centroid is simply the reference vector of the paper itself. Finally, we define the distance between two clusters C and C'. Let n_C and $n_{C'}$ be the number of papers in each cluster, and let r_C and $r_{C'}$ be the centroids of the clusters. Then

$$distance(C, C') = \sqrt{\frac{n_C n_{C'}}{n_C + n_{C'}}}\,(1 - \cos(r_C, r_{C'})) \qquad [2]$$

The square root scaling factor is used to force smaller communities to merge together before larger ones (11). This particular scaling factor leads to well balanced merge trees.

Our distance measure is a form of bibliographic coupling (12). A prominent alternative is to use cocitation analysis (13). In cocitation, two papers are judged similar if they are both cited by another paper. This is a very useful similarity measure. However, for this measure to work properly, a certain time-lag is required in order for papers to build up a citation record. Because our objective is to detect changes as early as possible, we opted for the common reference set approach. This also allows us to group papers that are not cited at all or only rarely cited, which is a significant portion of all papers.

To verify that the clustering algorithm and distance function were satisfactory, we compared the quality of the clusters we obtained to clusters obtained by standard techniques such as k means. One method of comparison is to count the number of journals and conferences needed to cover 90% of the papers in a cluster. The assumption here is that most journals, with a few exceptions, such as *SIAM Review*, are on a focused topic. Thus, the fewer the number of journals needed to cover a cluster, the tighter the cluster. In our tests, we found that the clusters obtained with the agglomerative algorithm were better defined than the clusters obtained by other methods.

Instabilities. To determine the set of natural communities, we examine changes in the agglomerative clustering trees under minor perturbations of the input data (14). More specifically, we compare different clusterings of the CiteSeer data, where we remove a small randomly selected set of papers (5%) before each clustering run. Given a base tree T_1, we compare how well the clusters in T_1 match with those in a second tree T_2, obtained on a different clustering run.

Let C and C' be two clusters of papers we wish to compare. Treating C and C' as sets, we define a value match(C, C') (between 0 and 1), as follows:

$$\text{match}(C, C') = \min\left(\frac{|C \cap C'|}{|C|}, \frac{|C \cap C'|}{|C'|}\right). \qquad [3]$$

The definition ensures that a high match values (close to 1) occurs when two clusters have many papers in common and are roughly of the same size. We define the value best-match (C, T) as the highest match (C, C') value for any cluster C' in T.

We considered a total of 45 clusterings of the CiteSeer graph. Each run uses a graph with a random 5% of the core papers removed. Full clustering runs on a data set of this size require

Table 1. Statistics of CiteSeer citation graph

Data set	*n*
Nodes	1,859,659
Nodes core	252,493
Edges	4,584,756
Average out-degree core	18
Median out-degree core	14
Median in-degree core	0

Table 2. Best-match values

Size range	No. of clusters in base tree	Average best-match	SD
100–400	2,812	0.42	0.07
401–1600	558	0.41	0.07
1,601–6,400	149	0.38	0.07
6,401–102,400	46	0.40	0.08

Table 3. Natural communities

Size range	No. of natural communities	Average best match	SD
100–400	116	0.74	0.05
401–1600	32	0.62	0.07
1,601–6,400	17	0.60	0.06
6,401–102,400	5	0.60	0.10

an efficient algorithm, and so we carefully exploit the sparseness of the underlying network. This allows for an efficient update of the set of intercluster distances after each merge. Code and data are available on request.

Table 2 gives the average best-match values of the clusters in the base tree T_1 matched against the other 44 trees. So, for example, the first row in the table shows that T_1 has 2,812 clusters containing between 100 and 400 papers. For each cluster C in this size range, we found the best matching cluster and its best-match value in trees T_2 through T_{45}. The average over these best-match values is 0.42 with a standard deviation of 0.07.

Table 2 shows that the average cluster matches quite poorly to its closest match in the other tree (average best-match value only $\approx$0.40). Interestingly, we can take advantage of these instabilities, because these clusters are not uniformly unstable and therefore can be exploited to uncover the true hidden structure of the data. In fact, a careful examination of the results of many runs shows that a small number of clusters, $\approx$170, appear in a good fraction of clusterings, and it is these clusters that correspond to recognizable topics. These "fixed points" in the CiteSeer graph are what we call natural communities, and these are the communities whose evolution we will track over time.

Natural Communities. We define natural communities as follows. We fix an input data perturbation value of 5%. Then we produce a series of subgraphs $G_1, G_2, \ldots, G_n$ of the original network G (the CiteSeer citation graph) where each G_i is the subgraph of G induced by a random subset of 95% of the core vertices of G. Our clustering algorithm then produces a set of clustering trees $\mathbf{T} = \{T_1, T_2, \ldots, T_n\}$. We choose the first tree T_1 as our base tree. We now define a natural community or cluster as follows.

Definition 3.1. A community C in base tree T_1 is natural *iff* in a fraction f of the clustering trees in $\mathbf{T}$ the best-match of C has a value greater than p, a predefined threshold.

The definition has two parameters: f, the fraction of trees out of n trees total, and p, a lower-bound for the best match. Depending on what values one chooses for these parameters, one obtains more or less well defined natural communities. In practice, we set these values sufficiently high to select clusters that are clearly different from the average cluster in the tree.

Using $n = 45, f = 0.6, p = 0.7$ for clusters with $<$1,000 papers and $p = 0.5$ for larger clusters, we found 170 natural communities of size 100 or greater in the CiteSeer graph, covering all aspects of computer science and portions of other fields like math and physics (see Table 3). These natural communities were selected from $>$3,500 clusters with size $>$100 in the base tree. Note that these natural communities vary in strength and their precise number depends, of course, on the setting of f and p. By using keyword data and journal titles, we found the natural communities to be quite coherent. In particular, the smaller to medium natural communities (up to a few thousand papers) correspond to well defined areas. Some example communities are listed in Table 4 (more details below). [Smaller natural communities are better defined. By using a different, but somewhat nonstandard, distance measure, one can also obtain better defined larger natural communities; ref. 14.]

We now turn to our main objective: the use of these natural communities in tracking the temporal evolution of the network.

Tracking Natural Communities

The key question remaining is how well the natural communities allow us to track the temporal evolution of the community structure in our network data.

In particular, we need to validate that when the network evolves over time and a few years of papers are added (*i*) there is not a dramatic shift in terms of natural communities, and (*ii*) that the occurring changes have a plausible interpretation in terms of the evolution of the field. These are inherently empirical questions. The results discussed below will show that our notion of natural communities satisfies both criteria, thereby making the concept a good candidate for use in temporal tracking in large networked data sets.

Method. To study the temporal evolution process in detail, we will track changes for a subset of the natural community data described above. We use two snapshots: the time periods 1990–1998 (referred to as the 1998 data set) and 1990–2001 (referred to as the 2001 data set). As such, our goal is to study changes in community structure as they occurred during the three years from 1999 to 2001. [The core set of the CiteSeer data set consists of papers available in digital form on the web. The literature coverage of the earlier years of the collection is less complete because the fraction of papers available on the web was limited, but by the late 1990s, the coverage for computer science had become quite comprehensive.]

In the 2001 data set, there are $\approx$100 natural communities containing between 100 and 350 papers. To analyze temporal changes in detail, we considered a subset of 20% of these communities (18 total) for closer analysis. Our selection of communities was representative of the overall set of communities in terms of size and year distribution. The communities contained 3,200 papers total. Let P_{2001} be the set of these papers. We create a citation subgraph containing only papers from P_{2001} and the references in these papers. We also removed some

Table 4. Established natural communities

Topic	Size in 1998	Size in 2001	Percentage in 2001
Digital watermarking	97	172	35.5
Data mining and association rules	78	128	25.0
Game search trees and artificial intelligence	161	172	8.7
Network traffic control	237	258	8.5
Crash recovery for distributed systems	139	151	7.3
Asynchronous circuit designand verification	231	244	6.6
Synchronous and asynchronous systems	203	219	6.4
Complexity theory: enumerability and querying	78	84	6.0
Query optimization for parallel databases	119	125	4.0
Fractal image coding and compression	86	89	2.2

Table 5. Emerging natural communities

Community	Size	Percentage in 2001
1998		
Networking (two communities)	237 + 130	
Quantum complexity	96	
2001		
Ad hoc/wireless networks	130	49.2
Quantum computation	140	30.0
Subcommunities		
Quantum complexity	82	15.9
Quantum algorithms and communication	38	76.3

low-quality information: all core papers that reference fewer than five other papers and all noncore papers only referenced once. This reduced the size of the subgraph by ≈20%. We repeat this procedure to create our 1998 graph by starting with papers up to and including 1998 from the set P_{2001} (a subset of 2,791 papers).

Results. We determined the natural communities for each data set by considering 10 clusterings for each graph and by using $f = p = 0.8$ in our definition of natural communities. We considered all natural communities with at least 75 papers. We then compared the natural community trees for the 1998 and the 2001 data, by finding for each 1998 natural community the best matching natural community in our 2001 data and vice versa.

Our first observation is that most of the 1998 communities have a good match (at least 70%) with a 2001 community (and vice versa). (Note that the 2001 data set contains ≈13% more papers than the 1998 set, with some communities growing by >30%.) Also, the natural community tree structures largely match up. Based on the matching data and the trees, we classify the natural communities in the 2001 data set as either established or emerging.

The established communities are given in Table 4. The table gives the size of the communities in 1998 and 2001 and then the percentage of papers in the 2001 community that appeared after 1998 (indicating the growth rate). The topic of each community was determined by considering the most frequent content words in the titles of the papers in each community. The communities are sorted by growth rate. We see that the growth rates vary quite a bit: some communities are very active and growing fast, such as digital watermarking and data mining, but several other communities appear stagnant, such as fractal image coding and compression, and query optimization for parallel databases.

From the perspective of temporal evolution, the most interesting changes involve the emergence of new communities. We identified two emerging communities: ad hoc/wireless networks and quantum computing. (In ad hoc networks, one studies self-configuring, distributed networks, generally wireless.) See Table 5. These emerging communities are consistent with recent developments in the field.

Wireless networks. Our first example is the emergence of the wireless community. In 1998, we have two natural communities centered around "network systems" with 367 papers. These communities consist of a combination of optical networking, distributed computing, and crash recovery papers with some initial papers on ad hoc/wireless networks. However, at this time, there is no well defined community on ad hoc/wireless networks. However, there is a significant change in networking papers over the 1999–2001 period, as ≈60 papers on ad hoc/wireless networks are added to the database. As a result, we find that, in the 2001 data, the ad hoc/wireless papers form a distinct natural community consisting of ad hoc/wireless papers from the 1998 set with the post-1998 papers added. In the 2001 cluster tree, this new community merges in with the larger network community at a higher level.

Quantum computing. A second example is the emergence of the quantum algorithms and communication community within quantum computing. This is an example of a community that is branching out over time (i.e., it is an evolving community). In the 1998 set of natural communities, we find that there is a natural community of size 96 that contains papers on quantum computing and complexity theory. In the 2001 set, this community has grown to 140 papers. However, the 2001 clustering now reveals further substructure: there are two distinct subcommunities of the size 140 community: one on quantum complexity (size 82; fairly stable) and another, fast growing community of 38 papers (20 more papers merge in separately). After examining the titles, it is clear that most of these papers cover quantum algorithms and quantum communication, both very hot topics in the past few years. So, in 1998, the quantum community was mostly centered on one topic; in 2001, the community was branching and growing quickly (theory conference agendas actually reflect this). Given the recent explosion of work in the area of quantum computing, it is encouraging to see these developments reflected in our natural community data.

In summary, these examples show that our notion of natural communities provides a promising tool for studying the temporal evolution of linked networks.

Related Work

Early pioneering work on discovering scientific communities using reference linkage information was done by Small and colleagues (13, 15). More recently, the NEC CiteSeer group succeeded in identifying intellectual communities in the CiteSeer database by using new variants of cocitation analysis (16) and network flow methods (17). The main impetus for the recent renewed activity in this area comes from the increasing importance of large linked networks in general, not just networks based on citation data (e.g., ref. 18). Indeed, recent work (19) explores the dynamics of social networks by simultaneously analyzing coauthorship and citation networks. For future work, it would be interesting to consider the relationship between authorship and natural clusters of papers as we identified here.

A key aspect that distinguishes our work is the emphasis on the temporal evolution of the network. As a consequence, for example, cocitation is less useful as a similarity measure, because it takes time to build up a cocitation record. Similarly, the use of highly cited papers, as in ref. 16, to identify core communities, also has limitations when looking for the most recent changes in the network involving emerging communities, because again it takes time to build up a citation record. (Ref. 16 measures the activity level of established communities by considering growth rates.) A related question is whether so-called hubs and authorities, as introduced by Kleinberg (20), form quickly enough to track recent changes. In general, a more detailed comparison between our natural communities and communities identified by using these other approaches is needed.

Another aspect that differentiates our work is its focus on stability: to track clusters over time, it is important that the clustering hierarchy be relatively stable. In many other applications, what matters most is not stability but finding an organization of items that, on human inspection, is coherent. With respect to the CiteSeer document collection, this amounts to identifying key computer science topics, such as systems and databases using titles and abstracts (17). However, the importance of stability is gaining recognition. Refs. 21 and 22 analyzed the stability of the two popular link-based ranking algorithms, HITS (20) and PageRank (23). They point out that intuitively we would not want the rankings for a given query to change much if the base data set, for example, the World Wide Web, is altered

slightly. They go on to develop algorithms that stabilize the HITS rankings.

Conclusions

We have provided a framework for studying the temporal evolution of the community structure of large linked networks. The notion of natural communities can be used to identify a relatively stable core of a hierarchical agglomerative clustering. Our approach exploits the inherent instabilities in clusterings in high-dimensional spaces (24). The true structure in the data are revealed by averaging out the large number of "accidental" clusters that emerge in any single clustering run. In our experiments on the CiteSeer network, we showed how the natural communities can be used to study the evolution of the network by tracking established communities and uncovering new, emerging community structure. Our next step is to evaluate our approach on other evolving linked networks.

We thank Steve Lawrence for making the October 2001 snapshot of the NEC CiteSeer database available to us. We thank Richard Shiffrin and Katy Börner for valuable feedback on an earlier version of this paper. We also thank Justin Yang for assistance with the experiments. This work was supported in part by National Science Foundation CAREER Award IIS-9734128, an Alfred P. Sloan Research Fellowship, National Science Foundation Information Technology Research Grant IIS-0312910, and the Intelligent Information Systems Institute at Cornell University sponsored by Air Force Office of Scientific Research Grant F49620-01-1-0076.

1. Watts, D. J. & Strogatz, S. H. (1998) *Nature* **393,** 440–442.
2. Watts, D. (2003) *Six Degrees: The Science of a Connected Age* (Norton, New York).
3. Barabàsi, A.-L. (2002) *Linked: The New Science of Networks* (Perseus, New York).
4. Erdös, P. & Rényi, A. (1960) *Pupl. Math. Inst. Hungarian Acad. Sci.* **7,** 17–61.
5. Giles, C. L., Bollacker, K. D. & Lawrence, S. (1998) in *Proceedings of the International Conference on Digital Libraries*, eds. Witten, I., Akscyn, R. & Shipman, F. M. (Assoc. Comput. Machinery Press, New York), Vol. 3, pp. 89–98.
6. Cohen, W., Kautz, H. & McAllester, D. (2000) in *Proceedings of the Association of Computational Machinery Special Interest Groups Knowledge Discovery in Data and Data Mining*, eds. Ramakrishnan, R., Stolfo, S., Bayardo, R. & Parsa, I. (Assoc. Comput. Machinery Press, New York), Vol. 6, pp. 255–259.
7. Pasula, H., Marthi, B., Milch, B., Russell, S. & Shpitser, I. (2003) in *Advances in Neural Information Processing Systems*, eds. Becker, S., Thrun, S. & Obermayer, K. (MIT Press, Cambridge, MA), Vol. 15, pp. 1401–1408.
8. Adler, R. J., Feldman, R. E. & Taqqu, M., eds. (1998) *A Practical Guide to Heavy Tails* (Birkhäuser, Boston).
9. Salton, G. (1989) *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer* (Addison–Wesley, Boston).
10. Jain, A. K. & Dubes, R. C. (1998) *Algorithms for Clustering Data* (Prentice-Hall, Upper Saddle River, NJ).
11. Duda, R. O. & Hart, P. E. (1973) *Pattern Classification and Scene Analysis* (Wiley, New York).
12. Kessler, M. M. (1963) *Am. Document* **14,** 10–25.
13. Small, H. (1973) *J. Am. Soc. Info. Sci.* **24,** 265–269.
14. Hopcroft, J., Khan, O., Kulis, B. & Selman, B. (2003) in *Proceedings of the Association of Computational Machinery Special Interest Groups Knowledge Discovery in Data and Data Mining*, eds. Ramakrishnan, R., Stolfo, S., Bayardo, R. & Parsa, I. (Assoc. Comput. Machinery Press, New York), Vol. 9, pp. 541–546.
15. Small, H. & Griffith, B. C. (1974) *Sci. Stud.* **4,** 17–40.
16. Popescul, A., Flake, G., Lawrence, S., Ungar, L. & Giles, C. L. (2000) *Advances in Digital Libraries, ADL 2000* (IEEE, New York), pp. 173–182.
17. Flake, G. W., Lawrence, S. & Giles, C. L. (2000) in *Proceedings of the Association of Computational Machinery Special Interest Groups Knowledge Discovery in Data and Data Mining*, eds. Ramakrishnan, R., Stolfo, S., Bayardo, R. & Parsa, I. (Assoc. Comput. Machinery Press, New York), Vol. 6, pp. 255–259.
18. Gibson, D., Kleinberg, J. M. & Raghavan, P. (1998) *Proc. Hypertext 1998 Conf.* **9,** 225–234.
19. Börner, K., Maru, J. T. & Goldstone, R. L. (2004) *Proc. Natl. Acad. Sci. USA* **101,** 5266–5273.
20. Kleinberg, J. M. (1999) *J. Assoc. Comput. Machinery* **46,** 604–632.
21. Ng, A. Y., Zheng, A. X. & Jordan, M. (2001) *Proc. Int. Joint Conf. Artificial Intelligence* **17,** 903–910.
22. Ng, A. Y., Zheng, A. X. & Jordan, M. (2001) *Proc. Assoc. Comput. Machinery Spec. Interest Groups Inf. Retrieval Conf., New York* **24,** 258–266.
23. Page, L. & Brin, S. (1998) *Comput. Networks ISDN Syst.* **30,** 107–113.
24. Aggarwal, C. C., Hinneburg, A. & Keim, D. A. (2001) in *Lecture Notes in Computer Science*, eds. Van den Bussche, J. & Vianu, V. (Springer, Heidelberg), pp. 420–434.